Analyse vidéo des mouvement des animaux pour l'animation de créatures 3D

DEA Imagerie, Vision, Robotique 2002-2003

Rapport de stage de recherche

Laurent FAVREAU

sous l'encadrement de Lionel REVERET

mené au sein de l'UMR GRAVIR (CNRS, INPG, INRIA et UJF)

Projet EVASION

Table des matières

Chapitre 1

Avant-propos

1.1 Présentation de l'INRIA Rhône-Alpes

L'INRIA, Institut National de Recherche en Informatique et en Automatique, est un établissement public à caractère scientifique et technologique, placé sous la double tutelle du Ministère de la Recherche et du Ministère de l'Economie, des Finances et de l'Industrie.

Créée en décembre 1992, l'unité de recherche INRIA Rhône-Alpes regroupe plus de 400 personnes réparties sur trois sites : la Zirst de Meylan-Montbonnot, le campus universitaire de Grenoble et le site technopolitain de Lyon (dont Lyon-Gerland et le domaine scientifique de la Doua).

Quatre pôles de recherche

L'INRIA Rhône-Alpes mène ses activités en étroite collaboration avec les laboratoires de recherche publics et privés, nationaux et internationaux, et elle entretient des liens privilégiés avec l'institut d'Informatique et Mathématiques Appliquées de Grenoble (IMAG). Ces activités sont organisées autour de quatre pôles de recherche :

- Maîtriser les systèmes et réseaux informatiques
- Aider à la conception et à la création
- Percevoir, simuler et agir
- Modéliser les phénomènes complexes

Développement et transfert de technologies

Acteur de la recherche au sein d'une région qui a retenu le pôle du numérique comme l'un des deux grands axes de développement technologique, aux côtés des biotechnologies, l'INRIA Rhône-Alpes mène une politique volontariste de transfert vers le monde économique et social et participe activement à la création d'entreprises innovantes.

Diffusion des connaissances

L' INRIA Rhône-Alpes accueille plus de 130 doctorants, ingénieurs et stagiaires. Ses chercheurs participent à l'enseignement supérieur au sein des universités et grandes écoles de la région Rhône-Alpes (Institut National Polytechnique de Grenoble, université Joseph Fourier, université Pierre Mendès-France, université de Savoie, Ecole Nationale Supérieure de Lyon). Afin d'assurer la diffusion des connaissances et du savoir-faire, l'INRIA Rhône-Alpes entreprend continuellement des actions en accord avec les objectifs suivants :

- Formation par la recherche
- Information scientifique et technique
- Partenariats internationaux
- Organisation de séminaires et colloques

1.2 Présentation de l'équipe Evasion

L'équipe EVASION (Environnements Virtuels pour l'Animation et la Synthèse d'Images d'Objets Naturels) du laboratoire GRAVIR (CNRS, INPG, INRIA, UJF) a été créée au 1er janvier 2003. Elle regroupe cinq chercheurs ou enseignants-chercheurs permanents, onze étudiants en thèse et un ingénieur expert. Ses travaux de recherche sont dédiés à la modélisation, à l'animation, et à la visualisation d'objets et de phénomènes naturels.

Pour cela, deux grand axes de recherche sont privilégiés : D'une part le développement d'outils fondamentaux destinés à spécification de scènes et objets naturels complexes, à la mise au point modèles alternatifs pour la forme, le mouvement et l'apparence ainsi qu'à la conception d'algorithmes reposant sur un niveau de détail adaptatif pour gérer au mieux la complexité ; d'autre part la validation de ces outils sur des scènes naturelles spécifiques, qui vont du monde minéral (océan, ruisseaux, lave, avalanches, nuages) au monde animal (simulation d'organes, visages corps et chevelure d'un personnage, mouvements d'animaux), en passant par les scènes végétales (morphogenèse de plantes, prairies, arbres).

Chapitre 2

Introduction

2.1 Sujet, motivations

Le but de ce stage est d'utiliser la vidéo, et plus particulièrement les séquences issues de documentaires animaliers, comme matière première pour l'animation de créatures 3D. Il y a à cela plusieurs motivations.

Tout d'abord, la vidéo est la source d'information la plus accessible. En effet, la capture de mouvement (*motion capture*) classique, à base de marqueurs optiques ou électromagnétiques, est difficilement réalisable sur les animaux. Même si elle a été réalisée sur les éléphants ou sur les chevaux (Fig. 2.1), il est souvent difficile, voire dangereux, de les équiper de tels systèmes. De plus, les scènes se déroulent la plupart du temps en extérieur, dans des conditions aléatoires ou dans des lieux peu accessibles. Tout cela constitue un fort investissement en temps et en moyen.

De plus, dans une perspective d'animation, la vidéo est un indice visuel très pertinent et souvent utilisé en pratique par les animateurs. Cependant, malgré l'avancée récente des techniques de modélisation et d'animation, la plus grande partie du travail en synthèse d'image consiste toujours en un grand volume de travail manuel. Ainsi, la réalisation de certains films d'animation (Fig. 2.2) occupe quelque centaines de personnes pendant plusieurs années.

Notre but est de, tout en laissant à l'animateur la plus grande liberté de contrôle possible, automatiser le processus de création d'animations à partir de séquences vidéo. Nous voulons créer un modèle de mouvement qui, une fois construit, soit utilisable facilement pour l'animation. Par ailleurs, nous voulons aussi pouvoir l'utiliser pour capturer du mouvement à partir d'autres séquences de façon plus automatique.

Comme nous nous limitons à l'utilisation de vidéos monoculaires, la reconstruction d'informations tridimensionnelles à partir de ces images est un problème sous-contraint. Pour y rémedier, nous proposons d'utiliser un modèle *a priori* de positions et de vitesses pour régulariser les données. Ici, nous nous intéressons principalement à l'amorçage d'une base de données, c'est-à-dire à la construction d'un premier modèle. Ce modèle sera initialisé en choisissant automatiquement sur la vidéo un faible nombre d'images clés, et en placant à la main les poses 3D correspondantes. Ensuite, nous pouvons inférer l'information 3D en position et en vitesse sur toute la séquence par prédiction linéaire à partir des images en niveaux de gris. Dans ce cadre, le principal enjeu qui se pose est la sélection automatique des clés.

FIG. 2.1 – un exemple de capture de mouvement sur les animaux à l'aide de marqueurs optiques.

FIG. 2.2 – Le film d'animation 'l'âge de glace'

2.2 Contribution

Nous montrons comment, à partir d'une séquence vidéo représentant un mouvement cyclique dont on a sélectionné quelques images, construire un modèle linéaire de mouvement. Ce modèle linéaire peut ensuite être utilisé pour reproduire le mouvement entier avec le modèle 3D, moyennant la donnée des poses correspondantes aux images clés et une prédiction linéaire de la géométrie à partir des images en niveaux de gris. Dans cette optique, nous proposons et confrontons plusieurs méthodes de sélection automatique du nombre et de la nature des clés à interpoler.

Chapitre 3

Etat de l'art

La plupart des travaux effectués, que ce soit en animation ou en vision, se sont concentrés sur les humains [OSBH00, SBF00, Bre98, FB02, ST03]. Le traitement des animaux en est assez proche dans son cadre général : il consiste à prédire le mouvement d'une chaîne articulaire, le squelette, à partir de l'observation d'un mouvement de surface, celui du corps. Cependant nous disposons de beaucoup moins de connaissances intuitives sur les mouvements des animaux que sur ceux des humains. Il est donc plus important de disposer de bonnes données de référence, et de les traiter de façon efficace.

3.1 Analyse vidéo de mouvements

De nombreuses méthodes existent pour retrouver la géométrie de créatures articulées à partir de vidéos monoculaires [ST03, Bre98, SBF00, FB02]. Beaucoup d'entre elles utilisent un modèle 3D *a priori* de la créature concernée. Elles consistent alors à chercher, à chaque pas de temps, la configuration du modèle qui maximise la corrélation avec l'image correspondante à l'aide d'une méthode d'optimisation (Fig. 3.1).

Ces méthodes souffrent principalement de deux problèmes. Tout d'abord, la reconstruction d'information en trois dimensions à partir d'une image est un problème largement sous contraint, notamment à cause de nombreuses ambiguïtés et occlusions. Contrairement aux méthodes multivues qui sont relativement robustes, le suivi monoculaire est un problème mal conditionné [ST03], donc très sensible au bruit. Ensuite, le nombre de degrés de libertés du modèle est élevé, au moins une trentaine, et rend l'implémentation des méthodes d'optimisation délicate. Depuis quelques années, il est admis que, pour espérer obtenir un résultat acceptable même avec un bon algorithme d'optimisation, les espaces de recherche doivent être réduits et contraints au maximum [Bre98].

Pour réduire cette instabilité, il faut régulariser les données, en 2D ou en 3D.

3.1.1 Régularisation des données

Régularisation 2D

Un indice très utilisé pour régulariser les informations au niveau image est le **suivi de points** précis placés sur le sujet. Ces points peuvent être soit des **marqueurs optiques**, soit des **points d'intérêt** détectés, par exemple, avec un détecteur de Harris. La géométrie est alors reconstruite à partir d'une matrice contenant les points suivis en 2D (la matrice de suivi). La plupart de ces méthodes supposent des objets rigides. Celle de Tomasi et Kanade [TT92] est fondamentale. Sous projection

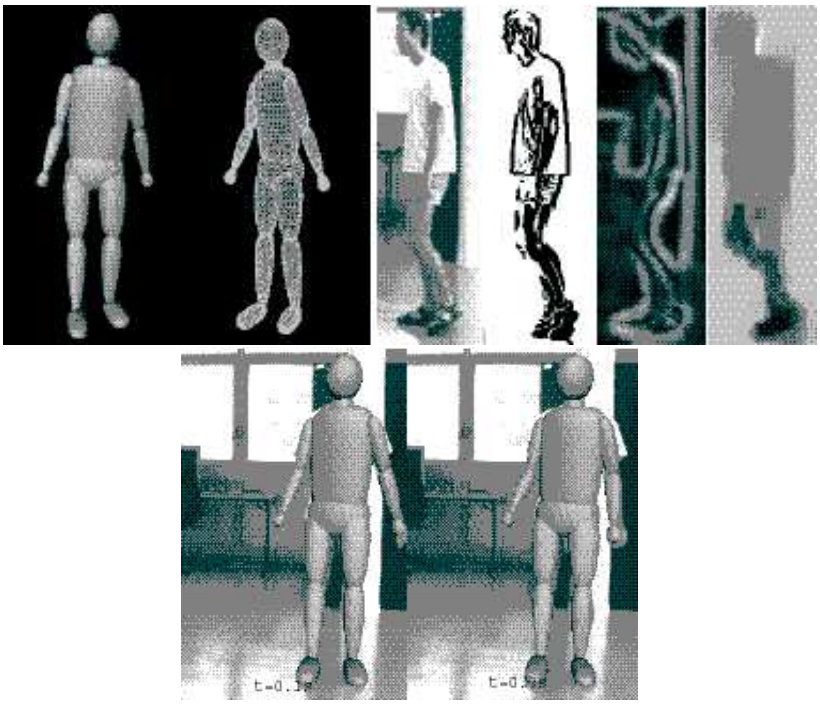

FIG. 3.1 – D'après Sminchiesescu et al.[ST03]. La configuration du modèle 3D est choisie pour maximiser la corrélation avec l'image.

orthographique, ils montrent que la matrice de suivi est de rang 3, et peut donc être factorisée en 2 matrices, qui décrivent respectivement la position de l'objet dans l'espace et sa forme en 3D. Cette méthode a beaucoup été reprise et améliorée, notamment par Seitz & Dyer [SD97] qui définissent une condition de rang pour que des points animés en 2D soient la projection affine d'un mouvement pseudo-cyclique. Récemment, certains travaux reconstruisent des objets non rigides, notamment des visages, avec un modèle 3D *a priori* [BV99].

Plus récemment, Bregler proposé une méthode [BHB99, TYAB01] qui factorise la matrice 2D en 3 matrices : une matrice de formes-clés, une matrice de poids qui définit la configuration de l'objet comme une somme pondérée des formes-clés, et une matrice de position (Fig.3.2). Cependant, cette méthode, qui prend en compte un minimum d'informations, est peu robuste. Elle convient bien à des applications telles que la reconnaissance d'activité, mais reste trop peu robuste pour l'animation.

Régularisation 3D

Il est aussi possible d'introduire des informations au niveau 3D. Pour le suivi de mouvements humains, Black et al. [OSBH00] utilisent des données de motion capture. Ces données sont d'abord découpées en cycles, puis une analyse en composantes principales (ACP) permet de construire une modèle de faible dimension pour décrire le mouvement en 3D (Fig.3.3).

Le **flot optique** est un outil souvent utilisé pour obtenir des informations dynamiques au niveau image. Il donne le vecteur vitesse correspondant à chaque pixel de l'écran [DJFJ00]. Dans le cadre du suivi, Black et al. [FB02] utilisent en entrée des données de *motion capture*. Ces données sont utilisées pour animer un modèle 3D, puis l'image de ce modèle est projetée sur plusieurs plans tout autour de l'objet. A partir de ces images, ils construisent des modèles de flot optique de faible dimension

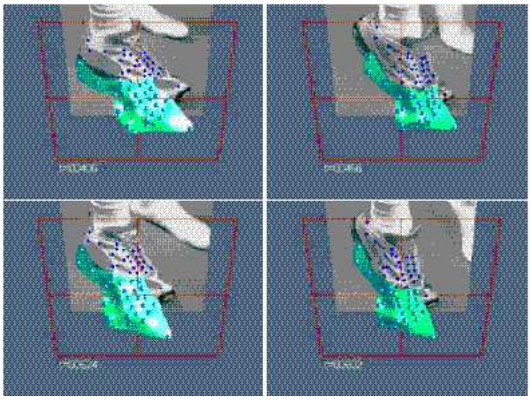

FIG. 3.2 – D'après Bregler[BHB99]

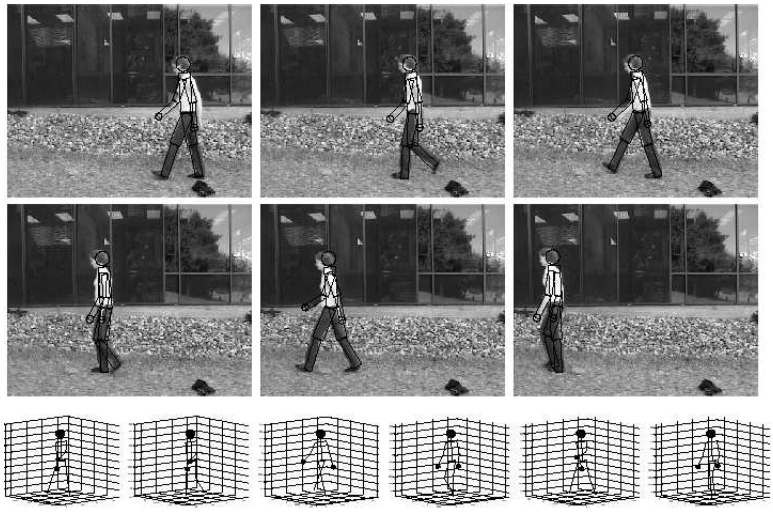

FIG. 3.3 – d'après Black et al.[OSBH00]. Un modéle dynamique *a priori* du modèle est appris à partir d'une capture de mouvements.

à l'aide d'une ACP. Ce modèle donne une information dynamique *a priori* sur le mouvement au niveau image.

Certaines méthodes, plus physiques, n'étudient plus le mouvement en terme de description temporelle (trajectoires), mais cherchent à le décrire par un ensemble de **paramètres physiques**. Un exemple simple est donné par Seitz et al [BSPK02] qui retrouvent les paramètres physiques d'un objet en chute libre. A l'aide d'une méthode d'optimisation et en connaissant la forme de l'objet et sa distribution de masse, ses paramètre extrinsèques (vitesse et position initiale), ainsi que la direction de la gravité sont extraits à partir d'une vidéo. Cette méthode donne un meilleur résultat qu'un suivi par filtre de Kalman.

Ces méthodes sont plus robustes que celles qui font uniquement appel à des informations en 2D, mais elles nécéssitent d'avoir des données *a priori* sur le mouvement de l'objet à suivre.

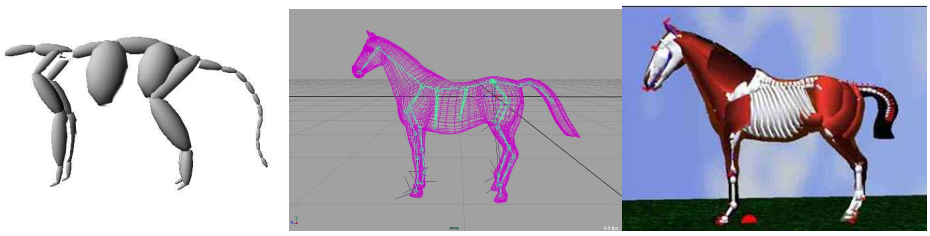

FIG. 3.4 – Un modèle à base d'ellipsoïdes, un modèle 3D classique de cheval, et un modèle de Wilhelms [WG97]

3.1.2 Le suivi dans un cadre bayésien

Le but du suivi est de déterminer le mouvement d'un modèle 3D à partir d'une séquence d'image. Pour cela, un cadre de travail bayésien est souvent mis en place [SBF00, OSBH00, FB02].

D'un point de vue probabiliste, le suivi consiste à déterminer, à l'instant t, la probabilité d'avoir une configuration donnée Φ_t du modèle 3D en connaissant l'image courante I_t . La base de ce formalisme est la loi de Bayes :

$$P(\Phi_t \mid I_t) = \frac{P(I_t \mid \Phi_t)p(\Phi_t)}{P(I_t)}$$

Bien entendu, la probabilité *a priori* $P(I_t)$ d'obtenir une image est indépendante de la configuration du modèle. Ainsi, la quantité cherchée est :

$$\Phi_t^* = argmax_{\Phi_t} P(\Phi_t \mid I_t) = argmax_{\Phi_t} P(I_t \mid \Phi_t)p(\Phi_t)$$

Autrement dit, la probabilité cherchée est proportionnelle à :

- La probabilité d'obtenir l'image en connaissant la configuration du modèle $P(I_t \mid \Phi_t)$. Cette probabilité peut être évaluée à l'aide d'un **modèle générateur** d'image et d'une **fonction de comparaison d'images**.
- La probabilité *a priori* d'obtenir la configuration $p(\Phi_t)$.

Dans notre cas, nous contribuons principalement à l'étude de la probabilité $p(\Phi_t)$.

3.2 Modélisation des animaux

Les animaux vertébrés, comme les humains, sont constitués d'une hiérarchie de segments articulés, le squelette, qui détermine leur structure, et de tissus qui entourent le squelette et leur donnent leur aspect extérieur. Pour les modéliser, la méthode la plus simple consiste à remplacer les joints par des ellipsoïdes. Plus précis est le skining classique, comme implémenté dans les logiciels graphiques industriels, où la peau est une surface 3D dont chaque point est influencé par le mouvement d'un ou plusieurs joints. Wilhelms [WG97] a construit des modèles plus réalistes comprenant os, muscles, tissus non musculaires et enfin peau, et où les muscles se déforment et se gonflent en fonction du mouvement des joints (Fig.3.4). Plus récemment [SWG02], elle a même proposé une méthode qui, à partir d'un modèle canonique de l'animal souhaité, et de la donnée d'une série de mesures, construit par morphing un modèle d'animal précis. Cependant le nombre de paramètres est encore trop important pour permettre une application stable en vision.

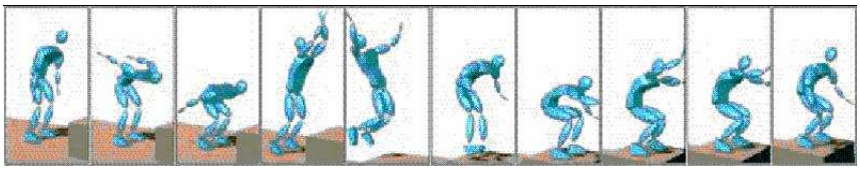

FIG. 3.5 – Une animation créée par la méthode de Popović [PW99]

3.3 Animation

Plusieurs techniques existent pour animer des modèles 3D. Certaines sont plutôt traditionnelles et massivement utilisées, d'autres sont plus avant-gardistes.

3.3.1 Animation par images-clés

L'animation par images clés consiste à créer une animation en ne donnant que les positions les plus importantes du mouvement, les **clés**, et en utilisant **une fonction d'interpolation** pour calculer les poses intermédiaires. Elle est énormément utilisée en pratique de part sa souplesse d'utilisation, et les outils récents permettent de choisir entre de nombreuses fonctions analytiques d'interpolation. Alternativement, Witkin et Popovic [WP95] interpolent à l'aide de courbes de *motion capture*, ce qui permet de synthétiser un mouvement relativement complexe avec peu de clés, moyennant une étape d'apprentissage.

Le principal problème inhérent à ce type de méthodes est la difficulté d'inclure des contraintes géométriques, comme par exemple 'le pied ne doit pas traverser le sol'.

3.3.2 Méthodes physiques

Une alternative est offerte par les méthodes physiques. Elles considèrent le mouvement non plus comme une succession d'état, mais dans son ensemble. Ainsi, alors que la modification d'une clé affectera seulement un voisinage de cette clé, ces méthodes essayent de trouver directement le mouvement qui satisfait le mieux les contraintes parmi l'espace de tous les mouvements possibles. Un tel système est constitué de trois éléments :

- une série de **contraintes**, soit des fonctions qui doivent garder un signe ou une valeur constants tout au long du mouvement.
- une **fonction de coût**, qui exprime combien une proposition est éloignée de la solution optimale.
- un **solveur**, souvent itératif, qui minimise la fonction de coût sous les contraintes.

Le choix de la fonction de coût varie selon l'objectif fixé. Ainsi, Popovic [PW99] cherche le mouvement le plus physiquement réaliste. Cette approche, qui est bien adaptée aux scènes très dynamiques (Fig. 3.5), donne de moins bons résultats avec des mouvements lents. De plus, la quantité de calculs nécessaire est très importante, aussi l'optimisation est-elle faite sur une version simplifiée du modèle.

A l'inverse, l'approche de Gleicher [Gle97] consiste à choisir le mouvement qui, tout en respectant les contraintes, reste le plus proche du mouvement d'origine spécifié par l'animateur. Cette méthode est plus simple à mettre en oeuvre et permet une édition interactive du mouvement, au détriment du réalisme. Plus récemment [Gle98], il a developpé une méthode spécialisée dans le retargetting de mouvement vers une créature dotée d'une structure similaire, mais possédant des joints de longueur différente.

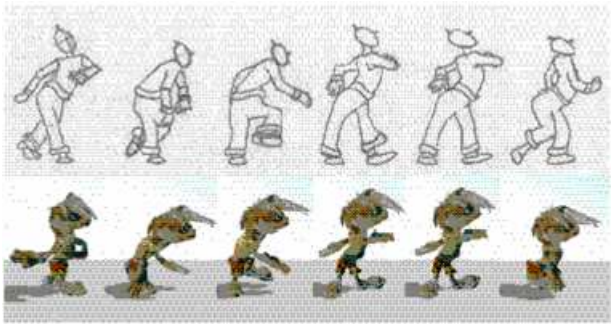

FIG. 3.6 – D'après Bregler et al. [BLCD02]

3.3.3 Animation à partir de vidéo

Récemment, Bregler et al.[BLCD02] ont mis au point une méthode pour créer des animations 3D à partir de cartoons. L'utilisateur choisit lui-même les images clés sur le cartoon, c'est-à-dire celles qui lui semblent décrire le mieux les position extrêmes du mouvement. Ensuite, il positionne à la main un modèle 3D pour avoir des poses correspondantes aux clés, puis du mouvement 3D est recrée par interpolation linéaire (Fig. 3.6). Cependant, une interpolation linéaire simple donne de nombreuses positions non valides. Il est donc nécessaire de générer un grand nombre de clés de façon semi-automatique puis de les corriger à la main. De plus, cette méthode ne refait que transposer un mouvement spécifique d'une créature vers une autre, sans portabilité. Néanmoins, notre inspiration se rapproche de cette méthode en ce qui concerne le passage de 2D à 3D.

Chapitre 4

Méthode proposée

4.1 Présentation de la méthode

Par rapport à la foule de méthodes existantes, notre cadre de travail apporte des difficultés particulières. Tout d'abord, dans le cas des animaux vertébrés, la structure qui contrôle le mouvement, c'est à dire le squelette, est enfouie dans les tissus. Les méthodes qui se limitent à une étude et une description surfacique du mouvement sont donc inadaptées. Au contraire, il nous faudra utiliser un **modèle 3D** de l'animal, avec muscles et squelette articulé.

Ensuite, le fait d'utiliser des vidéos monoculaires comme données d'entrée ne nous donne qu'une information 2D non calibrée, ce qui nous empêche d'utiliser les méthodes de vision les plus robustes. De plus, les données sont souvent bruitées, avec une fréquence d'échantillonnage limitée dans le temps (24 images par seconde) et dans l'espace (faible résolution). A cela s'ajoute l'éclairage naturel qui induit des variations importantes de luminosité et des ombres.

Enfin, la surface de nombreux animaux présente peu de texture, leur surface est non rigide, et les occlusions sont nombreuses. Dans ces conditions, il est difficile de traquer un point précis, et donc d'utiliser les méthodes à base de points d'intérêt. Comme notre but est l'animation, l'idée est d'utiliser les connaissances de l'utilisateur pour outrepasser ces limitations.

Pour commencer, nous segmentons l'image de l'animal du fond et nous réduisons la variation de luminosité. Après avoir effectué une réduction de dimension par ACP sur l'espace image, nous sélectionnons automatiquement quelques images clés, à partir desquelles nous construisons un modèle linéaire de prédiction du mouvement. Il nous reste ensuite à positionner à la main les poses du modèle 3D qui correspondent à ces images clés, et à utiliser le modèle linéaire précédent pour interpoler.

Nous construisons ainsi un modèle probabiliste statique et dynamique du système.

4.2 Construction du modèle

4.2.1 Acquisition des données

Sélection des données

Avant toute chose, nous avons constitué une base de donnée de vidéos. Ces vidéos sont sélectionnées afin de permettre une analyse facile :

- Des mouvements cycliques, selon un point de vue qui change peu.
- Des changements d'allures simples, par exemple une transition du trot au galop pour un cheval.

FIG. 4.1 – La segmentation par couleur. Les couleurs proches sont d'abord regroupées en paquets (*clusters*), puis la couleur de l'animal est séparée des autres.

- Des mouvements cycliques, avec un changement de point de vue simple, par exemple une rotation ou une translation.

Dans le cadre de ce stage, nous nous limiterons aux mouvements cycliques. Les vidéos que nous utiliserons principalement sont :

1. Une vidéo d'un guépard qui court, filmé de profil devant un fond mobile
2. Un deuxième guépard qui court, filmé de profil devant un fond mobile.
3. Un cerf au pas, filmé de profil devant un fond fixe.
4. Un cerf qui court, filmé selon un angle changeant devant un fond mobile.

Il nous faut maintenant extraire les informations importantes de ces vidéos.

Prétraitement

Pour commencer, l'image de l'animal doit être séparée du fond. Pour cela, nous pouvons procéder :

- Soit **par couleur,** en utilisant une distance euclidienne pour regrouper les couleurs proches, puis en séparant la couleur qui nous intéresse (Fig. 4.1). Cependant de nombreux animaux ont une couleur très proche de celle de leur environnement, ce qui rend l'efficacité de cette méthode aléatoire quand le fond et les objets sont proches.
- Soit **par soustraction de fond**. Dans le cas le plus simple le fond est fixe, et nous faisons une simple soustraction d'images. Si le fond est dynamique, le mouvement de la caméra est assimilé à une translation, et nous créons une mosaïque d'images pour reconstruire une grande image de fond (Fig.4.2).

Après la soustraction, l'image de l'animal est découpée et centrée autour du centre de gravité de l'animal. Afin d'éliminer les variations d'illumination, les images sont ensuite converties en noir et blanc. Enfin, nous les lissons avec un filtre gaussien. Nous obtenons les silhouettes des animaux (Fig.4.3).

Réduction paramétrique

Afin de réduire la dimension du signal, nous effectuons une analyse en composantes principales.

L'analyse en composantes principales, ou ACP, est une technique souvent utilisée pour construire des modèles linéaires de faible dimension à partir d'importants flots de données. Elle a pu s'appliquer par exemple à la création de modèles de flot optique pour décrire et analyser des mouvements complexes, comme les scènes naturelles [DJFJ00], ou les visages [TP91] .

Les données sont représentées par une matrice X, dont chaque ligne est le vecteur d'observation d'un échantillon. Dans un premier temps, les données sont centrées sur

FIG. 4.2 – Mosaique d'images

FIG. 4.3 – Quelques silhouettes d'animaux

la moyenne. Ensuite, on calcule les vecteurs propres E de la matrice de covariance X^tX, pour obtenir les principales directions de variation des données. Ces vecteurs sont orthonormaux et forment une base de X^tX. On a donc :

$$\begin{cases} E^tE = I \\ X^tXE = ED \end{cases}$$

Où D est la matrice diagonale composée des valeurs propres, ordonnées par ordre décroissant. Les vecteurs de E sont donc aussi trié par leur contribution au signal. Ensuite, en prenant un nombre réduit $m < n$ de vecteurs propres, une reconstruction approximative de x est donnée par :

$$x_{rec} = \sum_{i=1}^{m} (x_j.E_i)E_i = xE_rE_r^t$$

Le principal défaut de l'ACP est qu'elle repose sur une minimisation aux moindres carrés. En conséquence, un échantillon très bruité peut avoir un effet très répulsif, et éloigner le résultat trouvé de la bonne solution. Plusieurs travaux ont tenté d'y remédier. Ainsi, Xu et Yuille [LA95] rejettent entièrement les échantillons qui contiennent trop d'erreurs. Plus récemment, De la Torre et Black [lTB01, lT02] utilisent une norme robuste pour gérer les erreurs, ou *outliers*, au niveau des pixels. Nous appliquerons ici l'ACP sur des images en niveaux de gris, à la manière de [TP91], pour obtenir une paramétrisation réduite du signal image, tout en perdant le moins d'informations possible.

Dans notre cas, chaque ligne de la matrice d'observation est constituée des pixels de toute une image. Le nombre de composantes est choisi pour conserver 95% de l'énergie du signal de départ, en général 50 à 70. A ce stade, nous obtenons une série d'images, les composantes principales, et des trajectoires, qui ne sont autres que les projections du signal sur ces composantes principales (Fig.4.5)

Nous pourrions également faire un **et** logique entre la silhouette et l'image convertie en niveaux de gris, afin de conserver plus d'information sur la texture de l'animal. Nous avons choisi de ne pas le faire pour conserver un signal de plus petite dimension possible. En effet, le filtre gaussien sert à éliminer les hautes fréquences sur les bords de l'image, ce qui diminue le nombre de composantes nécessaires pour obtenir une bonne reconstruction. Les images en niveaux de gris, elles, augmentent ce nombre de composantes (Fig.4.4).

4.2.2 Construction du modèle linéaire de prédiction

Cohérence temporelle

En observant les trajectoires (Fig.4.5), il apparaît que les trajectoires les plus importantes sont clairement cycliques, elles codent donc une certaine cohérence temporelle du mouvement. Cependant, au fur et à mesure que leur importance diminue, les composantes semblent coder de plus en plus du bruit. Pour conserver le mieux possible les informations de cyclicité, les trajectoires obtenues sont filtrées grâce à une transformée de Fourier pour ne garder que les deux pics de fréquence les plus importants (Fig. 4.6). A présent, nous voulons construire un modèle pour prédire de la géométrie à partir d'images à partir de ces composantes.

Le modèle 3D utilisé

A partir de ces composantes, nous voulons prédire de la géométrie, c'est-à-dire le mouvement d'un modèle 3D qui représente une créature similaire à l'animal de la vidéo. Pour la vidéo du guépard, nous utilisons un modèle 3D de chat (Fig.4.10), et un modèle de cheval pour le cerf. Le modèle est constitué d'une peau (maillage 3D)

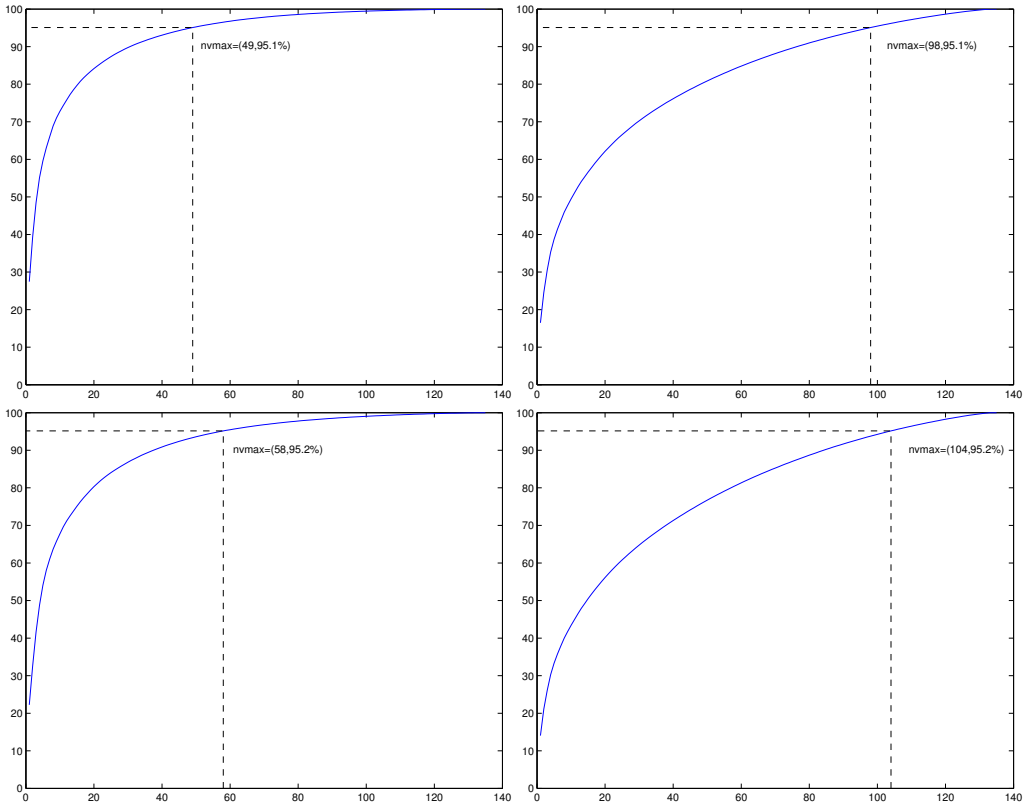

FIG. 4.4 – Le pourcentage du signal reconstruit en fonction du nombre de composantes principales utilisées, avec filtre gaussien (à gauche) ou sans filtre (à droite), et à partir des images en noir et blanc (en haut) ou en niveaux de gris (en bas). Le signal d'entrée est la vidéo du guépard après segmentation, soit 136 images de 94x40 pixels. Le meilleur résultat est obtenu avec des images en noir et blanc lissées.

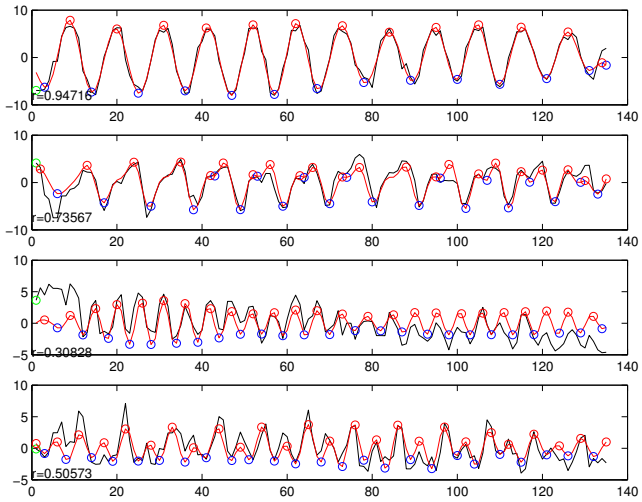

FIG. 4.5 – Les 4 premières composantes principales du guépard. Le signal d'origine est en noir, et le signal obtenu après filtrage par transformée de Fourier est en rouge. Les points rouges (resp. bleus) représentent les *maxima* (resp. *minima*) locaux. r est le pourcentage du signal conservé par le filtrage.

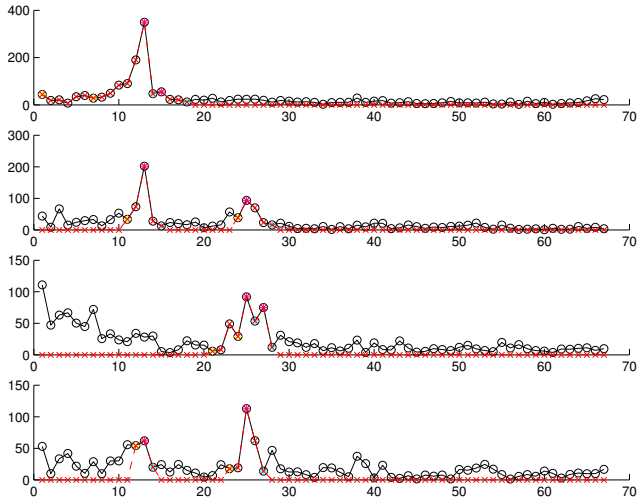

FIG. 4.6 – Les quatres premières composantes de la séquence du guépard, exprimées dans le domaine des fréquences. En noir : le signal d'origine. En rouge : le signal après filtrage. Nous conservons les deux pics de fréquence les plus importants.

et d'un squelette articulé, construit à partir de connaissance anatomiques (Fig.4.8). La racine de cette hiérarchie est située au niveau du bassin, ou *pelvis*. Ensuite, la peau est reliée au squelette par une technique de *skining* classique.

Formulation du modèle

Pour prédire le mouvement de ce modèle, nous voulons sélectionner quelques images-clés, les images qui décrivent le mieux le mouvement, placer les clés 3D qui correspondent à ces images, puis utiliser une interpolation linéaire pour reconstruire le mouvement dans son ensemble.

Supposons que nous avons sélectionné des images-clés sur la vidéo, et placé à la main les poses du modèle qui y correspondent. Formellement, soit X les clés sélectionnées sur la vidéo, et $Y_{nc\times nr}$ les clés étiquetées à la main[1]. Chaque ligne de X contient tous les pixels d' une image. Chaque ligne de Y contient les valeurs de toutes les rotations qui définissent une configuration du modèle 3D. Nous voulons trouver une application linéaire F d'interpolation entre les composantes principales et la géométrie, c'est-à-dire $F : XE \rightarrow Y$ qui vérifie :

$$F(x) = xEF = y$$

En sachant :

$$\forall i, F(XE)_i = Y_i$$

F peut être calculée par une formule de pseudo inverse :

$$F_{nv\times nr} = (XE)^+Y = ((XE)^tXE)^{-1}(XE)^tY$$

Ensuite, pour chaque image x, même si elle n'appartient pas aux clés, la géométrie prédite correspondante est donnée par :

$$y_{nc\times nr} = xEF = xE(XE)^+Y$$

Autrement dit, pour chaque image exprimée en terme de composantes principales, nous appliquons un traitement en deux étapes. Tout d'abord, nous la convertissons en termes de **combinaison d'images clés** avec $(XE)^+$. Ensuite, la géométrie correspondante est déduite par **interpolation** de nos clés en 3D.

Choix du nombre de composantes

En fonction du nombre de clés choisi, $(XE)^+$ a une nature différente :

- Si $nc < nv$, le problème est sous-contraint et donc numériquement instable.
- Si $nc = nv$, $(XE)^+$ est juste un changement de base
- Si $nc > nv$, les problème est surcontraint, et $(XE)^+$ est alors une solution aux moindres carrés.

Nous choisissons de prendre autant de clés que de composantes, afin d'avoir des espaces de même dimension. F est alors calculée par une simple formule d'inverse, et elle représente un changement de base de l'espace des composantes vers l'espace des clés.

Nous avons donc vu de façon formelle comment construire notre modèle linéaire de prédiction.Pour pouvoir construire effectivement ce modèle, il nous reste à savoir quelles clés choisir. Pour cela, nous proposons deux critères de sélection.

[1] nv est le nombre de composantes principales retenues, nc le nombre de clés, et nr le nombre de paramètres du modèle géométrique. En général $nr \simeq 30$.

4.3 Critères de sélection des clés à partir des images

4.3.1 La séquence de Muybridge

Afin d'avoir une référence solide, nous choisissons une courte séquence (20 images) de Muybridge, un photographe du 19ème siècle qui a réalisé de nombreuses études photographiques sur le mouvement humain et animal [Muy57] (Fig.4.9). Cette séquence présente deux avantages : tout d'abord, les images sont très précises, beaucoup plus que les vidéos dont nous disposons. Ensuite, comme la séquence est courte, il est possible de reconstruire la position en 3D du modèle pour toutes les images à la main. Cette séquence reconstruite nous permettra de quantifier l'erreur commise par les différentes méthodes de prédiction. La fonction d'erreur est définie par :

$$err(Y) = \sum_{i=1}^{ni} \sum_{j=1}^{nr} \left| Y_i^j - Yref_i^j \right|$$

où $Yref$ est la solution de référence .

Dans la suite de cette section, nous travaillerons sur cette séquence. Pour illustrer les variations d'erreur suivant les choix possibles de clés, nous avons calculé des statistiques (Fig. 4.7). Il apparaît que les écarts sont assez importants. Nous voulons donc trouver une méthode efficace, car les clés ne peuvent visiblement pas être choisies de façon aléatoire.

nb clés	min	max	écart-type	moyenne	cardinal
2	3.463493	8.460248	0.8421094	4.369701	66
3	2.433630	125.8865	9.755857	5.182684	220
4	2.363365	6798.872	362.6858	43.02484	495
5	1.996579	173.2116	7.788585	4.749238	792
6	1.897077	41.17268	1.833809	3.235762	924

FIG. 4.7 – Statistiques sur les clés. Pour chaque nombre de clés, la plus petite erreur de reconstruction possible, la plus grande, l'écart-type, la moyenne, et le nombre de combinaisons possibles.

FIG. 4.8 – Le squelette du chat

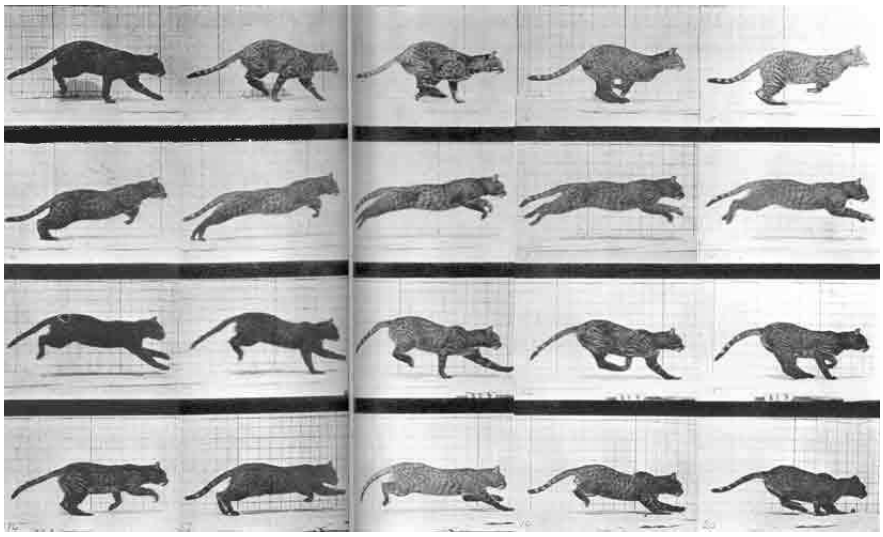

FIG. 4.9 – Une séquence issue des travaux de Muybridge

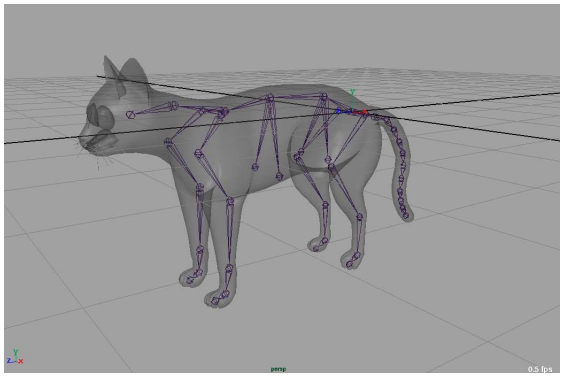

FIG. 4.10 – Le modèle 3D de chat avec son squelette

Choix du nombre de clés

Pour le vérifier, nous avons calculé, pour chaque nombre de clés possible ($1 \leq nc \leq 12$), la combinaison qui donne la plus faible erreur de reconstruction(Fig.4.11). Nous voyons que l'erreur diminue bien avec le nombre de clés, et que l'erreur est nulle lorsqu'on tente de prédire 12 images avec 12 clés. Le comportement de l'erreur est donc cohérent.

Bien sur, en pratique, un grand nombre de clés augmente la quantité de travail manuel à effectuer. Il faudra donc trouver un compromis entre quantité de travail et qualité de la reconstruction. La décroissance est quasiment linéaire, donc les choix du nombre de clés est assez libre. Dans la suite, nous utiliserons 4 clés.

4.3.2 Critère basé sur le conditionnement

Notre modèle linéaire est construit à partir d'une inverse sur les composantes principales au niveau image. On peur donc penser à choisir les clés qui minimisent le **conditionnement** de la matrice XE, ce qui peut garantir une reconstruction efficace. Le conditionnement d'un matrice est un réel qui caractérise la stabilité

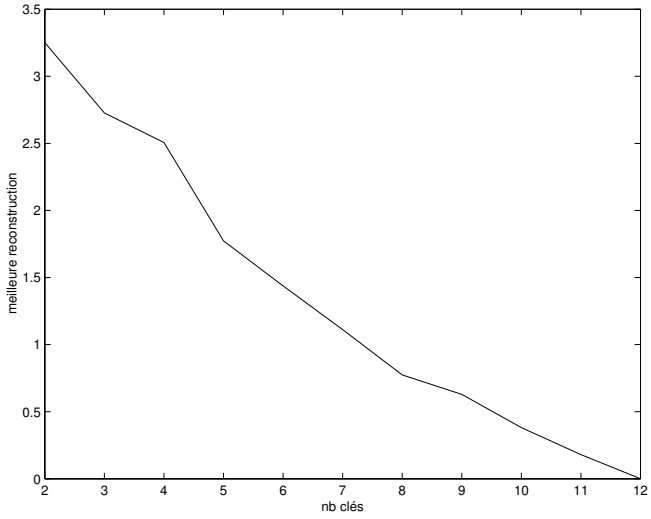

FIG. 4.11 – l'erreur de reconstruction diminue avec le nombre de clés.

clé 1	clé 2	clé 3	clé 4	conditionnement	erreur (degrés)
1	4	7	10	2.509804	2.363365
1	4	9	12	4.362513	2.396854
1	4	6	10	2.740954	2.400688
1	7	10	12	1.912157	2.414802
1	6	10	12	1.898749	2.439895
1	4	8	10	2.207094	2.470100
1	4	8	11	2.868717	2.480011
1	6	9	12	1.929530	2.488786
1	4	6	9	2.049284	2.557213
1	4	9	11	3.059900	2.562223

TAB. 4.1 – les clés qui donnent la plus petite erreur de reconstruction.

d'une matrice par rapport à l'inversion. La définition du conditionnement que nous utilisons est le rapport de la plus grande valeur singulière de la matrice par la plus petite. Si le conditionnement est proche de 1, la matrice est proche de l'identité, et donc stable relativement à l'inversion. A l'inverse, une matrice non inversible aura un conditionnement infini.

Afin de valider ce critère, nous avons comparé les combinaisons de clés qui donnent la plus faible erreur de reconstruction (Fig.4.1) à celles qui donnent le plus petit conditionnement (Fig. 4.2). Nous avons également calculé les résultats les plus mauvais selon ces deux critères (Fig. 4.3, 4.4). Il apparaît que ces deux critères sont très proches : Les meilleurs résultats, comme les pires, sont très proches dans les deux cas. Ce critère est donc un bon indice de la qualité de reconstruction. De plus il peut être calculé à partir des seules images, et ne nécessite pas de connaître explicitement le modèle 3D. Cependant, il demande toujours un calcul exhaustif sur toutes les combinaisons de clés possibles.

4.3.3 Critère basé sur les point extrémaux

A vu des trajectoires (Fig. 4.12), nous voyons que les composantes principales codent une certaine cohérence dans la mouvement . Nous proposons donc de cher-

clé 1	clé 2	clé 3	clé 4	conditionnement	erreur (degrés)
1	6	10	12	1.898749	2.439895
1	7	10	12	1.912157	2.414802
5	6	10	12	1.925487	2.710186
1	6	9	12	1.929530	2.488786
5	7	10	12	1.963815	2.738313
5	6	9	12	1.980576	2.975969
3	5	6	12	2.045020	3.271088
1	4	6	9	2.049284	2.557213
1	7	9	12	2.061095	2.684294
1	4	7	9	2.131017	2.605051

TAB. 4.2 – Les clés qui donnent le meilleur conditionnement.

clé 1	clé 2	clé 3	clé 4	conditionnement	erreur (degrés)
6	7	8	10	20711.035282	6798.872016
3	4	5	9	8154.001825	3957.920661
1	3	4	9	2848.152480	1445.092588
4	6	7	9	1847.007050	586.603184
6	7	8	9	1649.618950	541.113202
3	6	7	9	1612.805929	504.757437
6	7	8	12	1453.906893	434.447393
6	7	9	10	1108.107128	396.732882
6	7	9	12	1060.327741	359.591532
6	7	9	11	1003.280586	338.184908

TAB. 4.3 – Les clés qui donnent la plus grande erreur de reconstruction.

clé 1	clé 2	clé 3	clé 4	conditionnement	erreur (degrés)
6	7	8	10	20711.035282	6798.872016
3	4	5	9	8154.001825	3957.920661
1	3	4	9	2848.152480	1445.092588
4	6	7	9	1847.007050	586.603184
6	7	8	9	1649.618950	541.113202
3	6	7	9	1612.805929	504.757437
6	7	8	12	1453.906893	434.447393
6	7	9	10	1108.107128	396.732882
6	7	9	12	1060.327741	359.591532
3	4	6	7	1017.182365	276.353041

TAB. 4.4 – Les clés qui donnent le plus mauvais conditionnement.

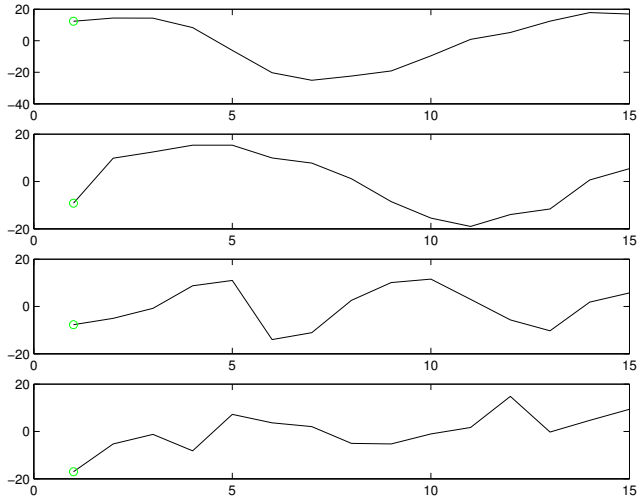

FIG. 4.12 – Les quatres premières composantes principales du chat.

cher les positions importantes du mouvement en cherchant les point extrémaux des courbes. Pour cela, nous considérons toutes les courbes en même temps. Nous ajustons des splines sur ces courbes, de manière itérative : A chaque étape, nous ajoutons un point de contrôle sur les splines, en minimisant l'erreur en position et en vitesse. Pour les quatre courbes, c'est le même point qui est ajouté (Fig 4.13).

Pour chaque nombre de clés possible, nous avons comparé le résultat donnée par cette méthode à celui donnée par le conditionnement, et à la solution optimale (Fig. 4.14). Il apparaît que ce critère donne de bons résultats, mais seulement pour un faible nombre de clés. Au delà de 6 clés, il devient moins efficace. Cependant, les autres méthodes nécessitent un calcul exhaustif sur toutes les combinaisons de clés, ce qui devient impossible à calculer pour de longues séquences. Pour conclure, le critère sur les points extrémaux est une bonne approximation, rapide à calculer, surtout pour de longues séquences.

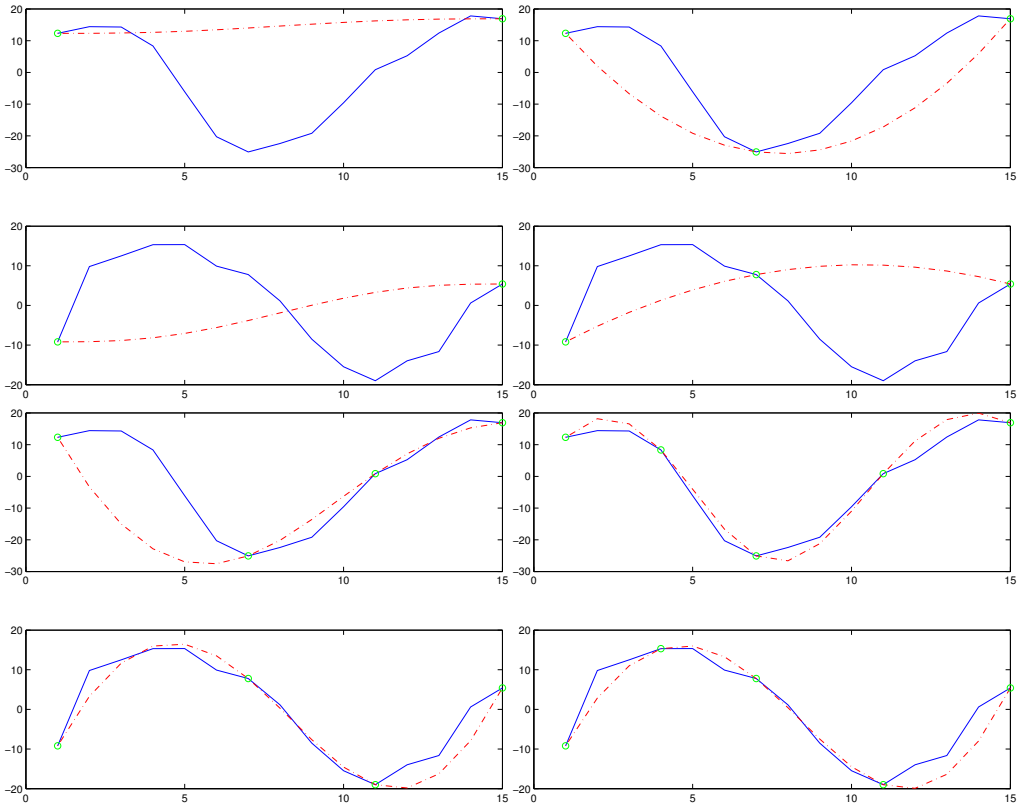

FIG. 4.13 – L'ajustement de spline : étapes 1,2,3 et 4

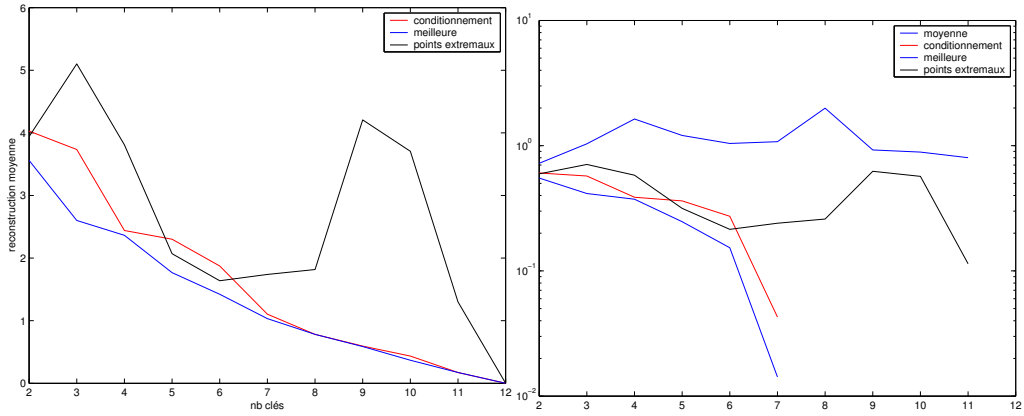

FIG. 4.14 – La sélection des clés par le conditionnement donne un résultat très proche de la solution optimale (A droite, avec la moyenne et une échelle logarithmique). Les points extrémaux donnent eux aussi un bon résultat si le nombre de clés est peu élevé.

Chapitre 5

Résultats

5.1 Reconstruction de la séquence de Muybridge

Pour la séquence du chat de Muybridge, le conditionnement (Fig. 5.2) comme le clustering (Fig. 5.3) donnent un bon résultat, proche de la solution de référence (Fig. 5.1).

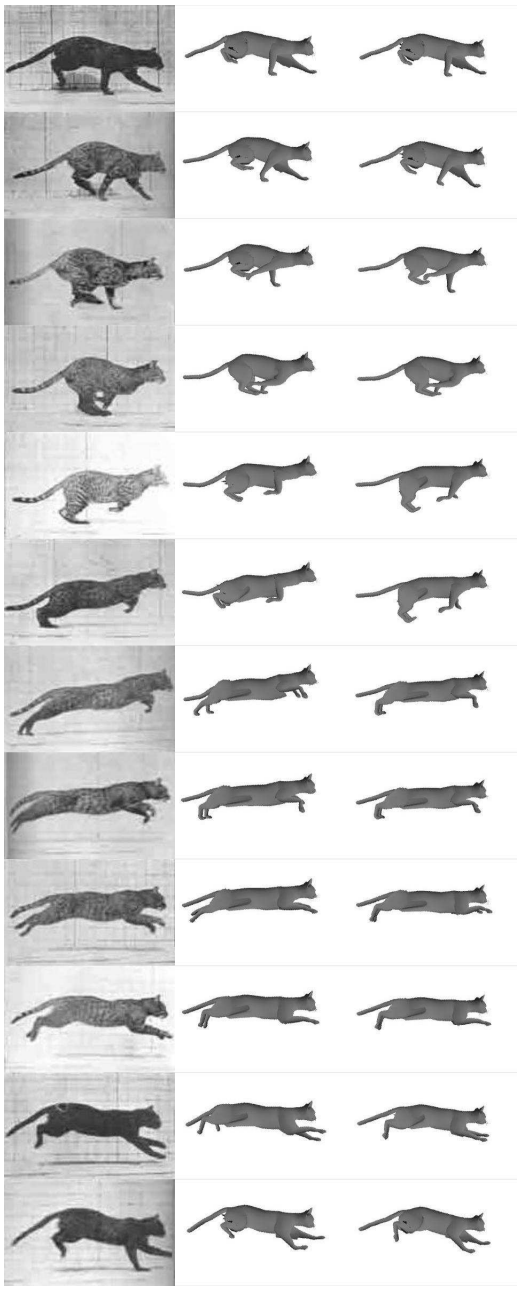

FIG. 5.1 – La reconstruction géométrique optimale de la séquence de Muybridge. Les clés sont les images numéro 1,4,7 et 10.

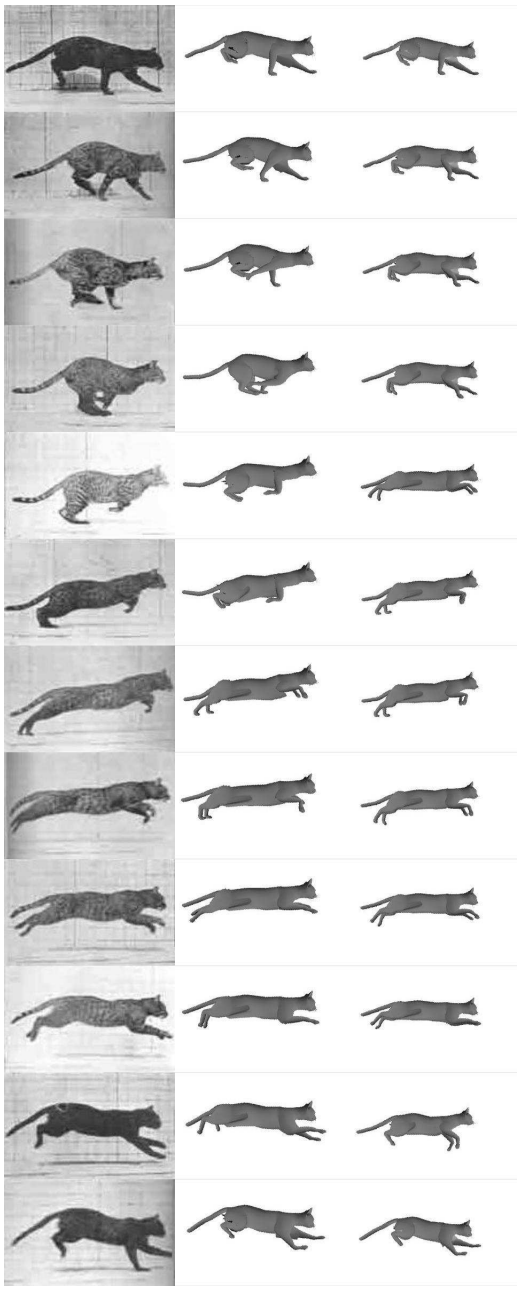

FIG. 5.2 – Prédiction à partir du conditionnement. Les clés sont les images numéro 1,6,10 et 12.

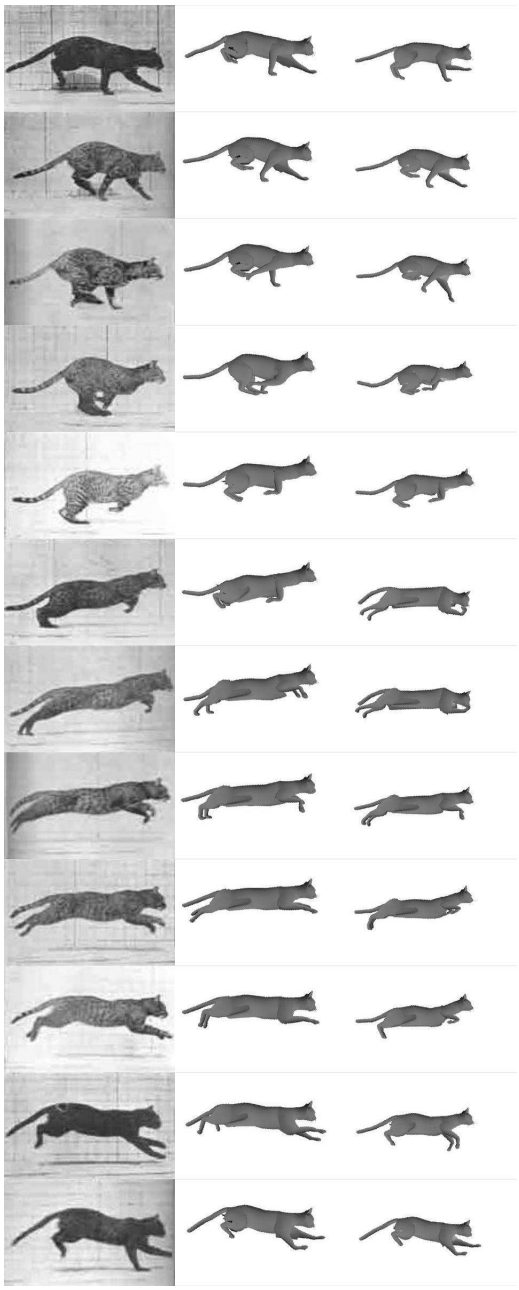

FIG. 5.3 – Prédiction à partir de l'ajustement de spline. Les clés sont les images numéro 2,5,8 et 12

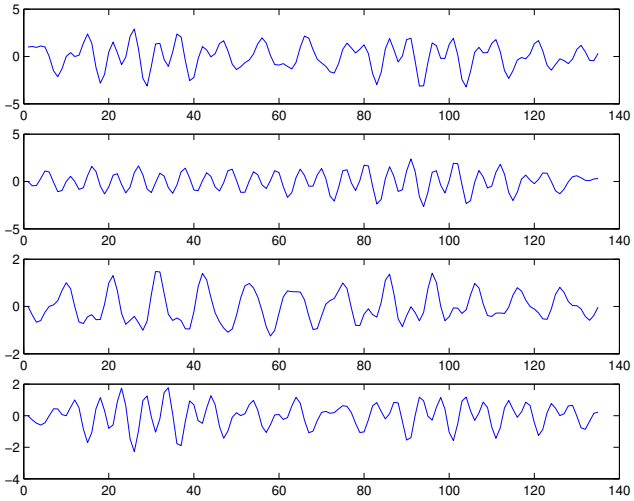

FIG. 5.4 – Les nouvelles trajectoires du guépard, dans l'espace des clés.

5.2 Reconstruction de l'animation du guépard

Nous avons utilisé notre méthode pour reconstruire une séquence vidéo plus longue, celle du guépard, longue de 136 images. Une fois les clés choisies, le modèle d'interpolation nous donne de nouvelles trajectoires, cette fois non plus en fonction des composantes principales, mais en fonction des clés (Fig.5.4), après application du changement de base des composantes principales vers les clés. Nous avons donc une description beaucoup plus intuitive du mouvement.

Nous commencons par effectuer l'ajustement de spline. Comme il y a plusieurs cycles d'animation, nous effectuons un clustering statistique sur les résultats (Fig. 5.5), afin de trouver les points vraiment importants pour décrire le mouvement. Nous regroupons les points proches en classes (*clusters*), puis, pour chaque classe, nous conservons l'individu le plus proche du centroïde. Les quatre clés trouvées (Fig. 5.6) correspondent bien à des poses distinctes et bien réparties le long du mouvement.

Comme il y a plusieurs cycles d'animation, nous pouvons calculer des statistiques sur les cycles. Sur la première composante principale (la plus cyclique), nous repérons les maxima locaux comme étant les débuts des cycles (Fig. 5.7). Les cycles sont ensuite découpés, puis normalisés afin de tous avoir la même longueur. A chaque instant t, nous calculons les vitesses $V_t = \Phi_{t+1} - \Phi_t$. Ensuite, nous pouvons calculer :

- Le cycle moyen des positions et des vitesses (Fig. 5.8 et 5.9).
- La distribution des positions et des vitesses $P(X_t)$,$P(V_t)$.
- La distribution dynamique du système : $P(V_t|V_{t-1})$.

5.3 Reconstruction de la séquence du cerf

Pour la séquence du cerf, il subsiste de nombreuse ambiguïtés qui nous empêechent de capturer le bon mouvement (Fig. 5.10). Comme nous n'utilisons que des silhouettes et que le mouvement du cerf est constitué de deux phases totalement symétriques [Muy57], le modèle ne peut pas faire la différence entre les pattes droites et gauches. Les clés du cerf sont visibles sur la figure (Fig. 5.11).

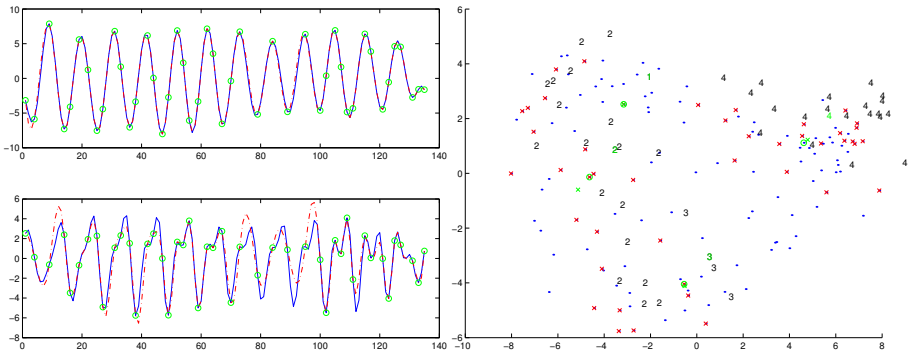

FIG. 5.5 – Ajustement de spline et clustering.
A gauche : nous ajustons une courbe spline sur les deux premières trajectoires, et nous ne gardons que les points de contrôle de la spline (en vert). Ces points décrivent les points importants du signal.
A droite : Ces points sont ensuite placés sur un plan (la position selon la première composante en abscisse, celle selon la deuxième composante en ordonnée). Le clustering nous donne les positions caractéristiques du mouvement, donc les images-clés.

FIG. 5.6 – Les quatre clés trouvées pour la séquence du guépard.

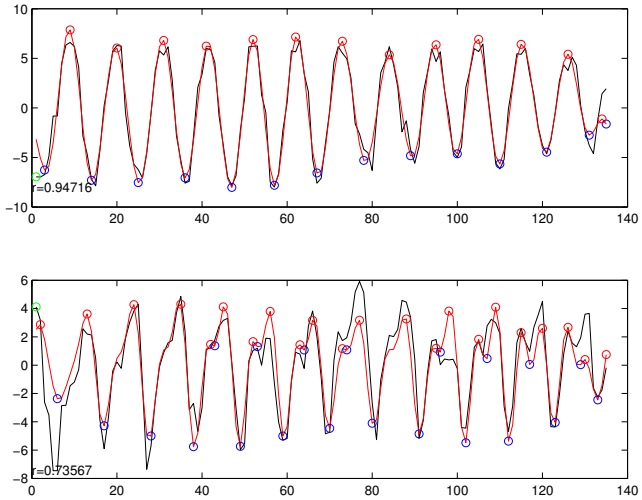

FIG. 5.7 – Les extrema locaux (points rouges) sur la première composante principale nous donnent les débuts des cycles.

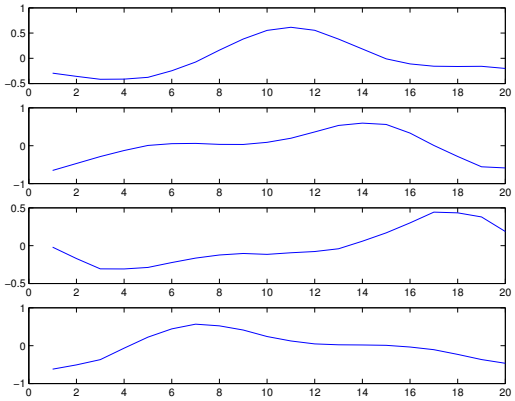

FIG. 5.8 – Le cycle moyen des positions retrouvé, en fonction des quatre clés.

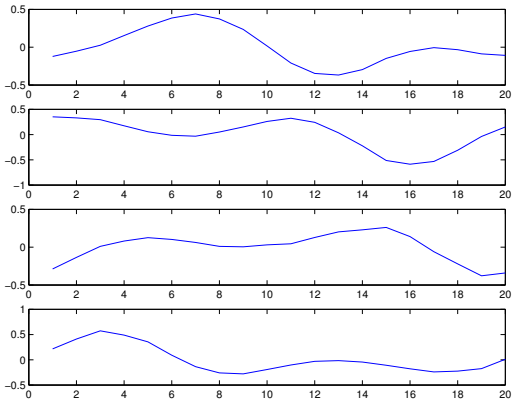

FIG. 5.9 – Le cycle moyen des vitesses.

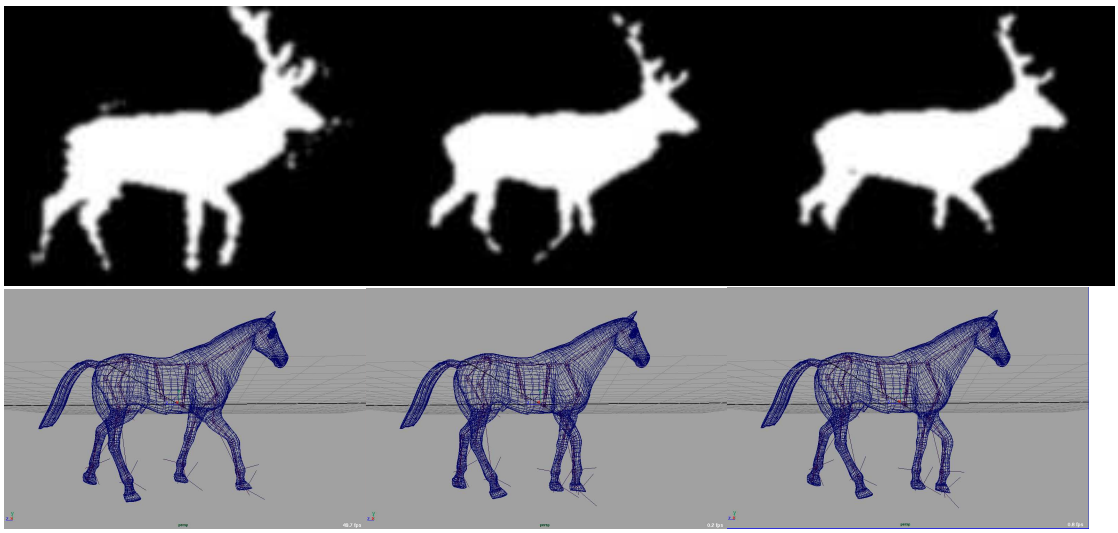

FIG. 5.10 – Prédiction du mouvement du cheval. A cause des ambiguïtés dues aux silhouettes, le bon mouvement n'est pas capturé.

FIG. 5.11 – Les quatres clés trouvées (par ajustement de spline) pour le mouvement du cerf.

5.4 Export en OpenGL

Jusqu'à présent, les modèles 3D utilisés étaient des modèle Maya. Afin de pouvoir suivre de nouvelles séquences, il est indispensable de pouvoir mettre en correspondance une image avec une configuration du modèle. Afin de pouvoir visualiser facilement notre modèle et d'avoir un modèle générateur d'image rapide, nous avons exporté le modèle en OpenGL. Plusieurs niveaux de détails sont possibles (Fig.5.14) :

- Remplacer les joints par des ellipsoides.
- Récupérer de façon plus précise en récupérant depuis Maya les morceaux de peau joint par joint. Pour chaque joint du squelette, nous récupérons les points de la peau qu'il influence le plus. Ensuite, nous récupérons le maillage formé par ces points, puis nous le convertissons dans le repère local du joint. Nous obtenons ainsi une liste d'affichage (*display list*) par joint. A l'affichage, nous parcourons la hiérarchie de repère induite par la structure de l'animal et nous appelons la liste correspondante à chaque joint depuis son repère local.

Nous pouvons à présent visualiser notre modèle (Fig. 5.12). Toutes les configurations du mouvement peuvent être décrites avec 10 paramètres : les six degrés de liberté du pelvis dans le repère monde, et le poids des quatre clés (Fig. 5.13).

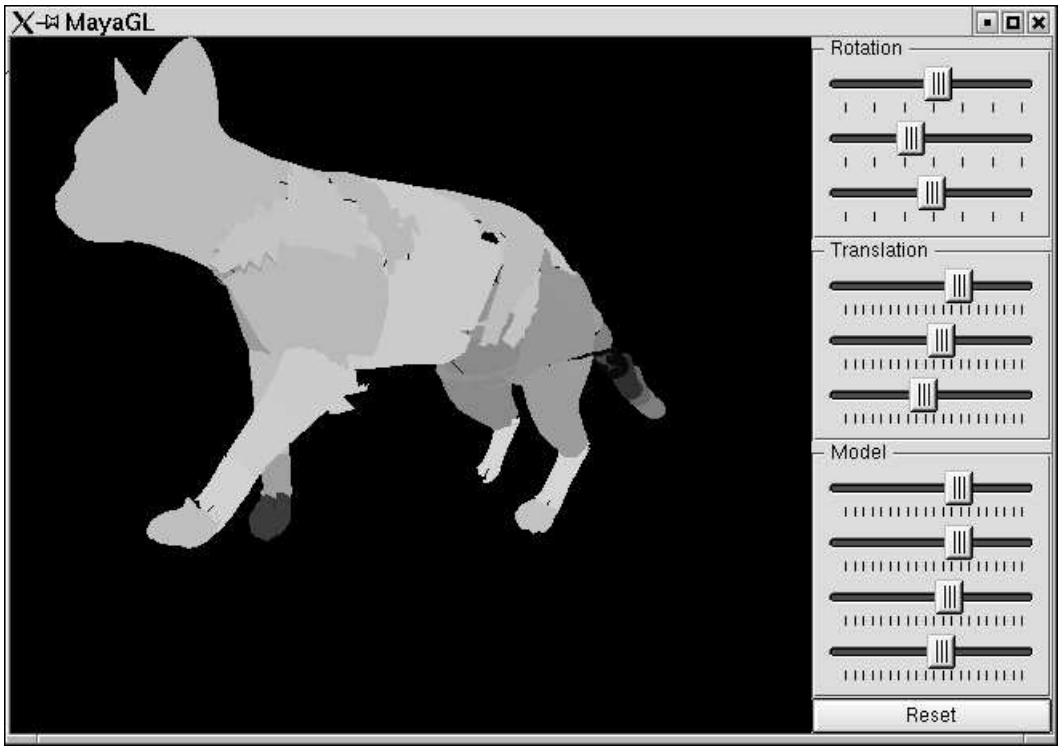

FIG. 5.12 – La visualisation en OpenGL du modèle. Toutes les configurations du mouvement peuvent être décrites avec 10 paramètres

FIG. 5.13 – Les quatre clés du chat.

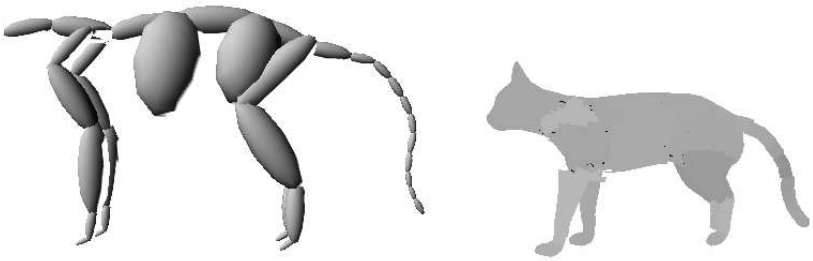

FIG. 5.14 – Le modèle en ellipsoïdes et avec les morceaux de maillage.

Chapitre 6

Discussion

6.1 Conclusion

A partir d'une séquence vidéo, nous avons montré comment choisir automatiquement des images-clés. A partir de ces clés, nous avons construit un modèle linéaire de prédiction de mouvement. Ensuite, moyennant la donnée des poses d'un modèle 3D correspondant aux images-clés, notre modèle peut alors être utilisé pour inférer une animation 3D complète .

Nous avons confronté plusieurs méthodes de sélection de ces clés : par conditionnement et par ajustement de spline. Sur une courte séquence de référence, nous les avons comparées avec une méthode optimale . La sélection par conditionnement donne un très bon indice de la qualité de reconstruction sans recourir au modèle 3D, mais elle nécéssite un calcul exhaustif. L'ajustement de spline donne une approximation légèrement moins bonne, mais très rapide à calculer. Sur de longues séquences, l'ajustement de spline est la seule méthode applicable en un temps raisonnable.

Nous avons également prédit une séquence cyclique d'une centaine d'images, comportant dix cycles d'animation. Nous avons ensuite construit un modèle statistique et dynamique du mouvement.

6.2 Perspectives

De nombreuses améliorations sont encore possibles.

En premier lieu, notre méthode demande à être testée et validée sur une plus large variété de séquences.

Ensuite, il nous reste encore à implémenter le suivi bayésien. Il nous faut donc décider d'une formulation du modèle dynamique.

Le fait d'utiliser une simple interpolation linéaire autorise le modèle à violer certaines contraintes physiques. En particulier, lors de la course, les pattes du chat passent à travers le sol. il faudrait inclure des contraintes physiques lors de l'élaboration du modèle.

De plus, le fait de n'utiliser que les silhouettes augmente beaucoup le risque de voir appraraitre des ambiguités. On le voit sur la séquence du cerf, ou l'analyse en composante principale capture un mauvais mouvement. Il y a plusieurs pistes pour y remédier :

- Utiliser plus d'informations sur l'image, notamment la texture ou la luminosité
- Surtout, utiliser un modèle plus dynamique pour représenter le mouvement.

Nous pourrions aussi chercher une meilleure mesure de la qualité de la reconstruction, par exemple en fonction de l'importance des joints. Ainsi, une rotation de

quelques degrés du pelvis aura un effet bien plus important qu'une rotation de même angle du petit doigt. De plus, le erreurs peuvent aussi bien s'ajouter que se compenser.

Le fait de traiter une séquence différente de la séquence d'apprentissage doit pouvoir nous permettre de corriger le modèle, en ajoutant des informations qui en étaient absentes, ou bien en les corrigeant.

Enfin, à plus long terme, le but est de considérer une succession de mouvement cycliques, avec les transitions entre les états, et bien sur de gérer des mouvements quelconques.

Bibliographie

[BHB99] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *CVPR99*, 16, 1999.

[BLCD02] Christoph Bregler, Lorie Loeb, Erika Chuang, and Hrishi Deshpande. Turning to the masters : Motion capturing cartoons. In *proceedings of SIGGRAPH 2002*, 2002.

[Bre98] C. Bregler. Tracking people with twists and exponential maps. In *CVPR98*, 1998.

[BSPK02] Kiran. S. Bhat, Steven. M. Seitz, Jovan Popovic, and Pradeep Khosla. Computing the physical parameters of rigid-body motion from video. In *Proc. 7th. European Conference on Computer Vision (ECCV), part I*, pages 551–566, 2002.

[BV99] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of SIGGRAPH 99*, pages 187–194, August 1999.

[DJFJ00] Yaser Yacoob David J. Fleet, Michael J. Black and Allan D. Jepson. Design and use of linear models for image motion analysis. In *International Journal of Computer Vision 36(3),171-193,2000*, 2000.

[FB02] Ronan Fablet and Michael J. Black. Automatic detection and tracking of human motion with a view-based representation. In *ECCV (1)*, pages 476–491, 2002.

[Gle97] M. Gleicher. Motion editing with spacetime constraints. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 139–ff. ACM Press, 1997.

[Gle98] Michael Gleicher. Retargetting motion to new characters. In *International Journal of Computer Graphics*, volume 32, pages 33–42, August 1998.

[LA95] L.Xu and A.Yuille. Robust principal component analysis by self-organizing rules based on statistical physics approach. In *IEEE Transactions on Neural Networks, 6(1) :131-143,1995*, 1995.

[lT02] Fernando De la Torre. A framework for robust subspace learning, 2002.

[lTB01] Fernando De la Torre and Michael Black. Robust principal component analysis for computer vision. In *ICCV*, volume I, pages 362–369, 2001.

[Muy57] Eadweard Muybridge. Animals in motion. Dover Publications inc., New York, 1957.

[OSBH00] Dirk Ormoneit, Hedvig Sidenbladh, Michael J. Black, and Trevor Hastie. Learning and tracking cyclic human motion. In *NIPS*, pages 894–900, 2000.

[PW99] Zoran Popović and Andrew Witkin. Physically based motion transformation. In *proceedings of SIGGRAPH 99*, pages 11–20, 1999.

[SBF00] Hedvig Sidenbladh, Michael J. Black, and David J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *ECCV (2)*, pages 702–718, 2000.

[SD97] S. Seitz and C. Dyer. View-invariant analysis of cyclic motion. In *International Journal of Computer Vision, 25 :1–25, 1997.*, 1997.

[ST03] Cristian Sminchisescu and Bill Triggs. Estimating articulated human motion with covariance scaled sampling, 2003.

[SWG02] Maryann Simmons, Jane Wilhelms, and Allen Van Gelder. Model-based reconstruction for creature animation. In *Proceedings of the ACM SIGGRAPH symposium on Computer animation*, pages 139–146. ACM Press, 2002.

[TP91] Matthew Turk and Alex Paul Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1) :71–86, 1991.

[TT92] C. Tomasi and T.Kanade. Shape and motion from image streams under orthography : a factorization method. *Int. J. of Computer Vision*, pages 9(2) :137–154, 1992.

[TYAB01] L. Torresani, D. Yang, E. Alexander, and C. Bregler. Tracking and modeling non-rigid objects with rank constraints. 2001.

[WG97] Jane Wilhelms and Allen Van Gelder. Anatomically based modeling. *International Journal of Computer Graphics*, 31(Annual Conference Series) :173–180, August 1997.

[WP95] Andrew Witkin and Zoran Popović. Motion warping. *Journal of Computer Graphics*, 29(Annual Conference Series) :105–108, 1995.